

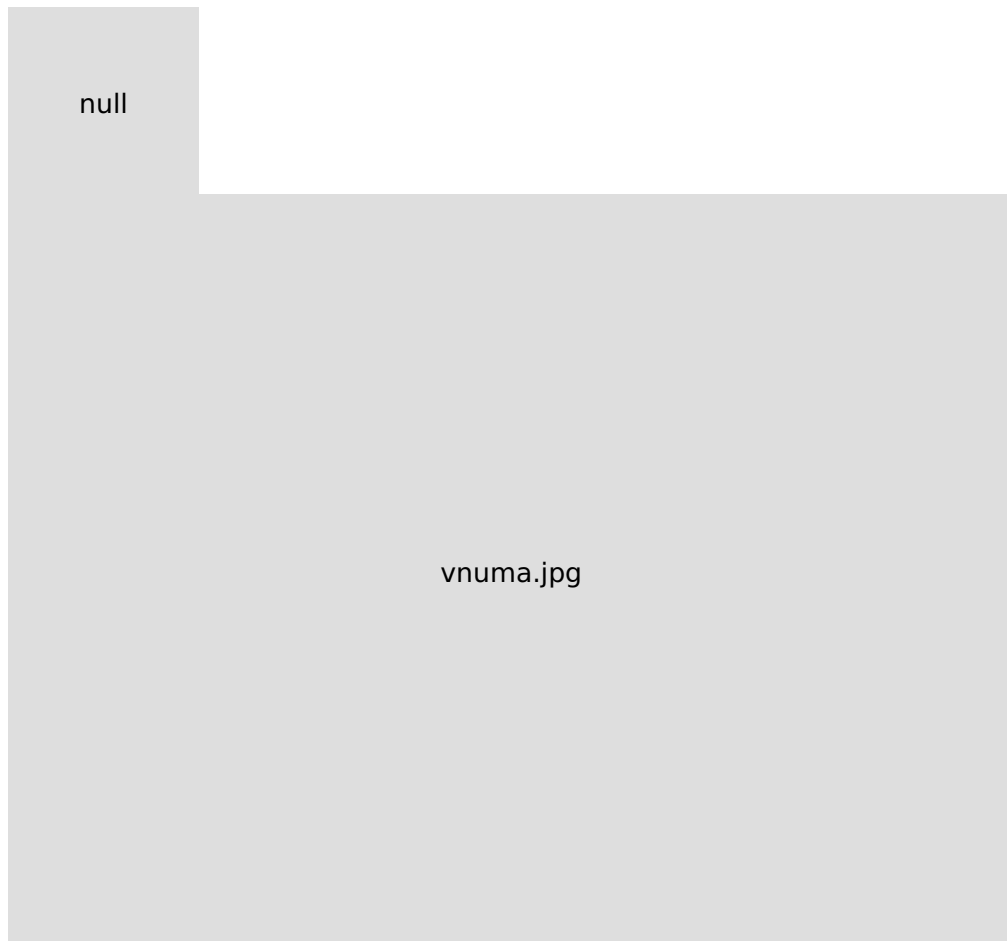
Using vnuma to Check Memory Usage and Non-Local Memory Access

vnuma is a NAS-designed tool that provides a user-friendly way for you to collect and visualize a PBS job's memory usage data as a function of time for the job.

The tool gathers data from the `/proc/meminfo` and `/proc/$pid/[stat,cmdline,numa_maps]` files, and provides information about the following types of memory usage:

- Total memory used on a node, and also the usage by all user processes
- Memory used by each individual user process
- Local and non-local memory access by each user process on a node

Non-local memory access lowers the performance of a process, which can cause the performance of the entire job to deteriorate. The following diagram shows an example of non-local memory access, where a process running on core 6 of socket 0 is accessing memory on socket 1:



You can run **vnuma** on NAS cluster systems, including Pleiades, Aitken, and Electra, but the tool is not supported on the shared-memory system Endeavour.

vnuma offers flexibility in its data collection and visualization functions. A brief description of its usage is provided in the next few sections. You can also jump to [Viewing vnuma Text Output](#) and [vnuma Options](#).

Collecting Memory Usage Data

You can use one of the following two methods to collect data using the **vnuma** command.

Method 1: Issuing the vnuma Command After a PBS Job is Submitted

After you submit a PBS job, you can issue the **vnuma** command from a PFE before or after the job starts running:

```
pfe% module load savors/2.x
pfe% qsub your_pbs_script
job_id
pfe% vnuma --save=directory_path [other options] job_id
```

Once the command is issued, it will not terminate even if you choose to exit the shell (unless the PFE where you issued the command is rebooted before your job starts running). **vnuma** checks whether the job is running, collects data, and exits when the job is completed. If the job stops unexpectedly before it completes, the data already collected is not lost. You can stop data collection by running:

```
pfe% vnuma --stop job_id
```

Note: You can view the job live without using the **--save** option. In this case, no data is saved for viewing afterwards.

Method 2: Issuing the vnuma Command Inside Your PBS Script

The following example demonstrates how to include the **vnuma** command inside your PBS script:

```
#PBS ..

module load savors/2.x
vnuma --save=directory_path [other options] $PBS_JOBID

mpiexec -np xx a.out

# You can stop data collection here if you do not want
# vnuma to collect data for the rest of the job, including
# usage by stream benchmark run by PBS epilogue as root

vnuma --stop $PBS_JOBID
```

The following tips and best practices apply to both data collection methods:

- The directory **directory_path** should be clean with no output from other jobs. If **directory_path** does not exist, it will be created.
- **directory_path** can be an absolute path or a relative path.
- For long running jobs, consider increasing the data collection period with the **--period** option. The default period is 1 second.
- For large jobs with many nodes, avoid using the **--all** option, which collects data from all nodes. The default is to collect data only from the head node. You can also use the **--hosts** option to collect data from a set of nodes. For example, **--hosts=0+2i** will collect data from nodes 0 (head node), 2, 4, ... from the **\$PBS_NODEFILE** of your job. If you know the nodenames before you issue the **vnuma** command, you can run (for example) **--hosts=r461i5n1,r463i2n9** to collect data from the nodes listed.

Viewing vnuma Output

You can analyze results by visualizing the output or by viewing the **vnuma** text output. Each method is described below.

Visualizing Memory Usage and Non-Local Memory Access

After you have collected the data, run the following commands:

```
pfe% module load savors/2.x
pfe% vnuma --load=directory_path [other options]
```

Use a maximum of two of the following options on the same plot. The default options are **--process** and **--ratio**.

- total**
shows the total memory used on a node, and also the usage by all user processes
- process**
shows the memory used (sum of local and non-local) by each individual user process
- ratio**
shows the ratio of non-local and local memory access by each individual user process. A ratio > 0 means that there is non-local memory access happening. A ratio of 1 means that there is equal amount of non-local and local memory access, which is detrimental


Please note the following tips and best practices:

- Use the **--geometry** option, or the **--smaxx** and **--smaxy** options, to set the size of the display window. For example, **--geometry=2000x1000** or **--smaxx=0.5 --smaxy=0.5**.
- You can exclude the display of some processes that use very little memory by using the **--min** option with the minimum size in megabytes (MB). For example: **--min=5**
- You can choose to include or exclude the display of certain processes with the **--include** or **--exclude** option. For example, **--include='34541|34542'** where 34541 and 34542 are the PIDs of two processes.
- Snapshots of the display can be automatically saved into postscript format with the **--snap** and **--snap-file** options.
- You can pause/unpause the live display with the 'z' key. Use the 't' key for stepping in time. Use Control-s to save a snapshot of the display. Unless **--snap-file** is specified, the default file name of the snapshot is **savors-snap.ps**.
- Once in a while, the **vnuma** command may not produce the display. Repeating the same command on the same window later or on a different window usually solves the problem.

Sample Plots

The following plot is generated by: **vnuma --load=. --total --process --min=20 --geometry=2000x1000**


The lines show the total memory usage (in MB) on the node by all processes (orange) and user processes (yellow). The dots show the per-process memory usage (in MB). Note that the scales for the lines and dots are different, as shown in the left and right axes. The large gap between the orange and yellow lines indicates a heavy use of buffer cache for I/O. The legend on the upper right provides the ID and command of each user process.



total-process_s1_28921.jpg

The next plot is generated by: `vnuma --load=. --include='34541|34542' --geometry=2000x1000`

This plot shows that the non-local memory access for process 34542 (in orange, where the ratio reaches about 0.5) is much more severe than for process 34541 (in red, where the ratio stays below 0.1). Dots are the memory used (sum of local and non-local) by each process (in MB). Lines show the ratio of non-local to local memory access. When a large ratio occurs, which can happen at the beginning of a job, it is important to check the memory usage. If the memory usage is very small, it is likely not causing any performance degradation.



34541-34542_s1_28900.jpg

Viewing vnuma Text Output

You can view the raw text data collected under the `directory_path` directory. The `time` file keeps parametersâ such as the time when `vnuma` starts collecting data for your job and the name of

the first node where data is collectedâ that are used internally by **vnuma**. For each node whose data is collected, there is a file called **out.rxxxixnx**.

A segment of a sample **vnuma** output file, **out.r431i4n1** is shown below. The definition of each column is shown on top for clarity. The second to last line lists the total memory used, based on the difference between MemTotal and MemFree in **/proc/meminfo**, on the node. The last line lists the total memory, summed over the local and non-local memory, used by all user processes.

```
time node pid psr socket local(MB) non-local(MB) command
1508445415 r431i4n1 34541 0 0 584.964 50.628 /u/.../overflowmpi
1508445415 r431i4n1 34542 1 0 451.176 194.56 /u/.../overflowmpi
1508445415 r431i4n1 34543 2 0 408.208 146.704 /u/.../overflowmpi
1508445415 r431i4n1 34544 3 0 410.096 145.752 /u/.../overflowmpi
1508445415 r431i4n1 34545 4 0 408.356 147.2 /u/.../overflowmpi
1508445415 r431i4n1 34546 5 0 407.348 186.04 /u/.../overflowmpi
1508445415 r431i4n1 34547 6 0 409.268 184.136 /u/.../overflowmpi
1508445415 r431i4n1 34548 7 0 408.736 165.908 /u/.../overflowmpi
1508445415 r431i4n1 34549 8 0 407.076 176.872 /u/.../overflowmpi
1508445415 r431i4n1 34550 9 0 408.312 184.616 /u/.../overflowmpi
1508445415 r431i4n1 34551 10 1 543.1 11.08 /u/.../overflowmpi
1508445415 r431i4n1 34552 11 1 549.036 11.772 /u/.../overflowmpi
1508445415 r431i4n1 34553 12 1 545.372 11.484 /u/.../overflowmpi
1508445415 r431i4n1 34554 13 1 559.884 10.416 /u/.../overflowmpi
1508445415 r431i4n1 34555 14 1 572.656 9.764 /u/.../overflowmpi
1508445415 r431i4n1 34556 15 1 579.368 9.212 /u/.../overflowmpi
1508445415 r431i4n1 34557 16 1 582.18 9.38 /u/.../overflowmpi
1508445415 r431i4n1 34558 17 1 582.98 10.264 /u/.../overflowmpi
1508445415 r431i4n1 34559 18 1 579.952 10.392 /u/.../overflowmpi
1508445415 r431i4n1 34560 19 1 593.952 9.608 /u/.../overflowmpi
1508445415 r431i4n1 - -1 45041.264 0 0 All Processes
1508445415 r431i4n1 - -1 11715.112 0 0 User Processes
```

vnuma Options

Run **vnuma -h** to see a list of command options:

```
pfe% vnuma -h
Usage: vnuma [OPTION]... [PBS.JOB_ID]

Options (defaults in brackets):
  --all                collect/show data from all job nodes
  --exclude=REGEX      exclude processes matching REGEX
  --geometry=GEOM      geometry of screen area to use
  -h, --help           help
  --hosts=LIST         collect/show data from comma-separated list of hosts
                      (I/Ni/I+Ni for Ith, every Nth, every Nth from I)
  --include=REGEX      include only processes matching REGEX
  --legend=REAL        fraction of width to use for legend [0.2]
  --legend-pt=INT      legend font point size [20]
  --load=DIR           load data from DIR instead of running job
  --min=INT            exclude process memory usage less than INT MB/s [0]
  --no-frame           do not show window manager frame
  --period=SECS        collect/show data every SECS seconds [1]
  --process            show memory usage per process
  --ratio              show ratio of non-local mem to local mem per process
  --save=DIR           collect job data to DIR without visualization
  --smaxx=REAL         max fraction of screen width to use [1]
  --smaxy=REAL         max fraction of screen height to use [1]
  --snap=PERIOD        take snapshot every PERIOD amount of time
                      (use suffix {m/h/d/w} for {mn/hr/dy/wk})
  --snap-file=FILE     save snapshots to FILE
  --stop              stop collecting data
  --total              show total system and total process memory usage
```

The vnuma tool was developed by NAS staff member Paul Kolano.

Article ID: 552

Last updated: 13 May, 2021

Revision: 57

Running Jobs with PBS -> Optimizing/Troubleshooting -> Managing Memory -> Using vnuma to Check Memory Usage and Non-Local Memory Access

<https://www.nas.nasa.gov/hecc/support/kb/entry/552/>